بسم الله الرحمن الرحيم

الجمهورية اليمنية- صنعاء
جامعة المستقبل- كلية علوم الحاسوب
قسم نظم المعلومات (IT)

درس هااااام جدا :
أدوات التحقق من المدخلات Validation Controls

في تقنيةASP.NET

أعداد المهنـــــدس/ نبيل محمد لطف مصلـــــي
Email:-nabil299@gmail.com
تحت اشراف الدكتور/ عمـــا ر الحرازي



مقدمة :

من اهم العمليات التى تحملها صفحات الويب على عاتقها هى استخلاص البيانات من المستخدم ليتم فيما بعد غالبا حفظها فى قاعدة بيانات .. ارتبطت عملية الحصول على البيانات من المستخدم بعملية التحقق من البيانات قبل حفظها فى قاعدة البيانات حتى نتجنب المشاكل فيما بعد.

من المفضل أن تتم عملية التحقق على جهاز المستخدم **Client-Side** عن طريق أكواد جافا سكريبت(**JScript**) حيث يتم إخبار المستخدم مباشرة بالخطأ قبل أن يتم عمل ال **PostBack** ..

البعض يتســـأل ماهي عـــمــلية PostBack ؟

والجواب:- هي عملية العودة الى السيرفر والرجوع الى المتصفح (المستخدم) أذا لم يحدث خطا في الصفحة.

لكن عملية التحقق لا يجب أن نكتفى بأن تقوم فى ال **Client-Side** فقط ..تخيل معى أن مستخدم الصفحة التى يجب أن نتحقق من البيانات عليها قبل حفظها لديه خلفية عن الويب وقام بحفظ نسخة من الصفحة على جهازة وقام بتحريرها وحذف منها كود الجافا سكريبت **JScript**الخاص بعميلة التحقق واستخدمها فيما بعد لإرسال بيانات مخالفة وخاطئة.هنا وجب وقوع عملية التحقق فى السيرفر دائما. **Server-Side Validation**

طبعا كتابة أكواد التحقق يدويا أمر متعب سواء كانت أكواد جافا سكريبت أم أكواد **ASP.NET** لذلك وفرت لنا ميكروسوفت مجموعة من الأدوات تسمى **Validation controls** والتى تضعها على الصفحة وتربطها بالأداة التى تريد إجراء التحقق عليها.

يمكنك بواسطة هذه الأدوات التأكد من وجود بيانات أو مقارنتها ببيانات أخرى أو حتى تقييدها بقواعد معينة.. كل هذه الأدوات تدعم التحقق على الخادم والتحقق على على السيرفر وذلك بشكل افتراضى.. واذا منع المستخدم متصفحة من تنفيذ أكواد الجافا سكريبت سيكون التحقق على السيرفر فقط.

Validation controls :

محموعة من الأدوات توجد فى التبويب **Validation tab** فى صندوق الأداوات **ToolBox** وعددها ستة أدوات . تستخدم للتحقق من مدخلات المستخدم كالتحقق من ان لا يترك الحقل فارغا .أو ان تكون القيمة عدد صحيح أو حتى اكبر من قيمة معينه..وإن فشلت العملية فإنها تمنع الصفحة من عمل **PostBack** وتظهر رسالة خطأ على أداة.

هذا ملخص عن الادوات إلى ان يأتى التفصيل لاحقا:

--الأداة : **RequiredFieldValidator** تتحقق أن أداة معينه تحتوى على قيمة عند عمل **Submit**للنموذج .فإذا كنات الأداة فارغة يظهر خطأ.

-الأداة : **RangeValidator** تتحقق من أن القيمة التى تحملها الأداة تقع فى مدى معين ..بين تاريخين ..رقمين ..أو حتى حروف..

: **CompareValidator** --تتحقق من أن القيمة التى تحملها الادة أكبر من ..أصغر من ..تساوى ..أكبر من وتساوى .. أصغر من وتساوى... قيمة ثابته تحددها أنت أو حتى قيمة تحملها أداة أخرى .. كما يمكنها أن تتأكد من أن الأداة تحمل نوع معين من البيانات أعداد صحيحة مثلا.

: **RegularExpressionValidator** --تتحقق من ان القيمة التى تحملها الأداة تتبع تعبيرا معينا تحدده أنت سابقا ..كأن تكون بريد إلكترونى مثلا او حتى رقم تليفون..

CustomValidator --تتيح لك إمكانية كتابة أكواد التحقق يدويا على السيرفر بواسطة أكواد **ASP.NET** وحتى على جهاز المستخدم بواسطة أكواد.. **JScript** وتلجأ لها فى حالة لم تفيدك الأدوات الأخرى.

: **ValidationSummary** --هذه الأداة ليست للتحقق من شىء معين ولكن يعرض عليها ملخص لكل عمليات التحقق الفاشلة.

يجدر الإشارة إلى إمكانية استخدام أكثر من أداة تحقق على أداة واحدة ..وكما ستعلم لاحقا أنك لو استخدمت الاداة **RangeValidator** مثلا مع **textbox** فإن عملية التحقق ستنجح لو كان ال **textbox**فارغا لأنه ببساطة لا توجد بيانات للتحقق منها.. وهنا تظهر الحاجة الى استخدام الأداة **RequeredFieldValidator** مع نفس صندوق النص للتأكد من عدم تركه فارغا.

لاحظ ان أكثر الأدوات التى تستخدم أدوات التحقق معها هى الأداة.. **textbox** كما انه يمكنك أن تستخدمها مع **DropDownList** وحتى **InputText** ولاحظ فى حالة الأداتين **DropDownList**و **InputText**وامثالها تتم عملية التحقق على الخاصية **value** على عكس ال **textbox**والتى تتم عملية التحقق فيه على الخاصية .. **text** لذلك إن حاولت إستخدام ادوات التحقق على **DropDownList** مثلا لا تنسى أن تعطى قيمة للخاصية **Value**لكل عنصر من عناصرها..

لاحظ أيضا أنه لايمكنك استخدام أدوات التحقق مع **RadioButton** أو **CheckBox** وهذا شىء منطقى وبديهى.

عملية التحقق: **The Validation Process**
يمكنك ان تقيم عملية التحقق بطريقتين طريقة اوتوماتيكية حيث تتم عملية التحقق عند الضغط على الزر**Submit** ليس الزر **Submit** فقط ولكن اى زر له الخاصية **CausesValidation**تساوى (**True** حيث أنه عند الضغط على أى زر فإنه تتم أولا مراجعة القيمة المسندة للخاصية **CauseValidation** فإذا كانت **false** تعمل الصفحة وكأنه لا توجد أدوات تحقق عليها واذا كانت **true** يتم التحقق من الأدوات على النموذج اذا حدث اى خطأ تظهر الرسائل للمستخدم.

لاحظ أن العمليات الأوتوماتيكية تعتمد على وقوع حدث النقر لزر له الخاصية CausesValidationتساوى true ولا تعتمد على حدوث postback لأى سبب آخر كحدث TextChanged لصندوق نص مثلا وهنا يمكنك الإعتماد على التحقق بطريقة يدوية وكتابة أكواد بناء على النواتج.

بعد المقدمة التى تعرفنا فيها على أهمية التحقق من مدخلات المستخدم ..وتعرفنا أيضا على وظيفة كل أداة من تلك الأدوات بشىء من الإيجاز الشديد .... دعنا نتعمق.

تشتق فئات الأدوات ValidationControls من الفئة BaseValidator والتى تحتوى على الأعضاء المشتركة لجميع الأدوات وبدراستها نوفر الكثير من التكرار..

من أهم أعضاء الفئة: BaseValidator

--الخصيصة : ControlToValidate حدد فيها اداة للتحقق منه البيانات بها.

الخصيصة:Display: تحدد كيف ستظهر رسالة الخطأ للمستخدم ولها ثلاث وضعيات

- : Staticوفى هذه الحالة سيتم حساب الفراغ اللازم لرسالة الخطأ وحجزة بحث تظهر الأدوات التى بجانبها لكن بعد تعدى الفراغ اللزم لها .
- : Dynamicفى هذه الحالة لن يتم عبور الجسر قبل الوصول إليه (مش هنقدر البلاء قبل وقوعه) بمعنى انه لن يتم حجز المساحة اللازمة لرسالة الخطأ ..لوكن عند وقوعها سيتم إزاحة الأدوات التى بجانبها بالمقدار اللازم لطول الرسالة. ربما تبدو هذه الوضعية مناسبة ..لكن احذر منها خاصة اذا كانت صفحتك مملوءة بجداول أو أدوات كثيرة.
- : Noneفى هذه الحالة لن تظهر رسالة خطأ أصلا ولكن سيتم إيقاف الصفحة من عمل !! Postback يعنى ستقوم الأداة بوظائفها من إيقاف الصفحة من عمل postbackوغيرها ولكن دون ظهور رسالة للمستخدم. وهذا عند التعامل مع الأداة ValidationSummaryلإظهار الأخطاء كلها جملة واحدة.

--الخصيصة : Enabled لتحدد إذا كانت الادة ستقوم بعملها كأداة تحقق أم أنها لن تعمل وتأخذ قيمة من اثنتين True أو. false

--الخصيصة : ErrorMessage تحدد فيها الرسالة التى ستظهر للمستخدم فى الأداة ValidationSummaryفى حالة فشل عملية التحقق .. وتقبل قيمة نصية. String

--الخصيصة : Text وتحدد فيها رسالة الخطأ التى ستظر للمستخدم على أداة التحقق نفسها فى حالة فشل عملية التحقق.

لاحظ الفرق بين الخصيصتين Text و(ErrorMessage)

--الطريقة (الوظيفة :()IsValid) تستخدم بشكل أساسى فى الكواليس للتحقق من كون العملية فشلت أم نجحت ..ربما تتساءل إذا كانت تستخدم بشكل أساسى خلف الكوالس ..لماذا لم

تجعلها ميكروسوف .. ()Privat Function هل أخطأت ميكروسوفت؟!
أكيد لم تخطئ ميكروسوفت ..تابع معى:
اذا كان لديك صفحة ويب تقوم بضرب القيمة فى صندوق النص فى العدد 10 وإظهار الناتج فى أداة Label .. هنا يجب التحقق من أن المستخدم يدخل قيم رقمية وإلا سيظهر خطأ أثناء الوقت التشغيل RunTime إذا ادخل قيمة نصية . فإذا أدخل المستخدم قيمة نصية وضغط على الزروكان التحقق فى جهة المستخدم Client-Side Validation متاح فلن يسمح المتصفح للمستخدم بعمل postback لتنفيذ الكود فى حدث النقر للزر. أما اذا كان التحقق فى جهة المستخدم غير متاح فإن الادة وللأسف ستسمح للزر بعمل postback وللأسف سيتم تنفيذ كود الضرب الموجود فى الزر مع اظهار رسالة الخطأ (ما فائدتها إذا..(! لذلك وبما أنك لا تضمن أن يكون التحقق فى جهة المستخدم متاح ستعتمد على التحقق فى جهة السيرفر وستقع فى المشكلة السابقة .. لذلك عليك أولا وقبل كتابة الكود الخاص بعملية الضرب أن تتحقق إذا كانت عملية التحقق التى تقوم بها الادة نجحت أم فشلت . لذلك ستختبر القيمة التى تعمود بها الوظيفة ()IsValid فإن كانت true ننفذ الكود وإن كانت false لا نريد منه أن ينفذ لنا شىء لذلك سيكون كود حدث النقر للزر يشبه التالى:

```
If Validator.IsValid Then
Label1.Text = TextBox1.Text * 10
EndIf
```

حيث Validator تشير إلى اسم الأداة البرمجى. ID

فى الحالات المشابهة :لن تختبر القيمة التى تعود بها الوظيفة IsValid() لكل أداة على حدى لذلك ستستخدم الوظيفة ()IsValid للصفحة والتى تعود بالقيمة true اذا لم يكن هناك أى عملية تحقق على الصفحة فاشلة وتعود بالقيمة false اذا كانت هنا على الأقل أداة تحقق فشلت عملية التحقق بها.

--هناك بعض الخصائص مثل ForeColor و BackColor و font وغيرها من الخصائص التى ليس لها علاقة مباشرة بأدوات التحقق ولكن لها علاقة بالأدوات بشكل عام . لذلك سنناقشها فى النهاية مع بعض الأشياء الأخرى تحت عنوان ((اللمسة الجمالية.))

أولا الأداة: RequiredFieldValidator

هى ابسط الأدوات تتحقق من أن الأداة التى ترتبط بها ليست فارغة أو تحتوى على مسافات فقط فى هذه الحالة تفشل العملية اما اذا كانت تحتوى بيانات فإن العملية تعبر بكل سلام.

وهناك خاصية لها تسمى InitialValue فإذا حددت لهذه الخاصية قيمة ..فإن عملية التحقق تفشل اذا ادخل المستخدم نفس البيانات اللتى حددتها انت فى الخاصية InitialValue وتنجح العملية اذا ادخل اى قيمة أخرى أو حتى ترك الحق فارغ.
وكمثال لهذه للاداة:-

```
<asp:TextBox runat="server" ID="Name" />
<asp:RequiredFieldValidator runat="server"
ControlToValidate="Name"
Display="dynamic">Name is required
</asp:RequiredFieldValidator>
```

حيث ستظهر هذه الجملة "**Name is required**" فى الأداة اذا عمل المستخدم **Submit** للفورم ولم يدخل قيمة فى صندوق النص.. **name**

وترى هذا المثال واضحا عند التسجيل فى المنتديات مثلا.. حيث حقل إسم المستخدم مثلا لا يجب أن يترك فارغا .

ثانيا الأداة: RangeValidator

تتحقق من أن القيمة التى تحملها الادة المرتبطة بها تقع بين مدى معين.
ومن أهم خواصها:
--الخصيصة **Type** والتى تحدد نوع البيانات التى ستتحقق الأداة من وقوعها بين مدى معين ويمكن أن تكون.**Currency, Date, Double, Integer, and String**

--الخصيصة **MinimumValue** تحدد أقل قيمة صالحة للبيانات داخل الأداة المرتبطة بها.
--الخصيصة **MaximumValue** تحدد أقصى قيمة صالحة للبيانات داخل الأداة المرتبطة بها.

وهذا مثال للأداة:

```
<asp:RangeValidator runat="server" Display="dynamic"
ControlToValidate="DayOff" Type="Date"
MinimumValue="08/05/2005 MaximumValue="08/20/2005>Day Off is not within
the valid range
</asp:RangeValidator>
```

حيث تتحقق الأداة من أن التاريخ المدخل الى الحقل **DayOff** يقع فى المدى من ٥ اغسطس ٢٠٠٥ إلى 20 اغسطس ٢٠٠٨ وإلا يظهر رسالة تدل على ان التاريخ غير مقبول ولا يقع ضمن المدى.

ثالثا الأداة: CompareValidator

تتحقق هذه الأداة من كون البيانات التى تحملها الأداة المرتبطة بها أكبر من ..أصغر من ..تساوى ..أكبر من وتساوى .. أصغر من وتساوى... قيمة ثابته تحددها أنت أو حتى قيمة تحملها أداة أخرى .. كما تتأكد من أن الأداة تحمل نوع معين من البيانات أعداد صحيحة مثلا.

وتستخدم عند التسجيل فى المنتديات للتحقق من ان حقل كلمة المرور وحقل التحقق من كلمة المرور يحملان نفس القيمة .... وايضا للتحقق من أن تاريخ الميلاد أقل من تاريخ التسجيل!!

من أهم خصائصها:

--كما فى أختها The RangeValidator لها الخصيصة Type والتى تحدد نوع البيانات التى ستتحقق الأداة من ادخالها فى الحقل.. كما تحدد نوع البيانات التى ستتم مقارنتها Currency, Date, Double, Integer, and String .

يا حبيبى.. قد تتساءل لم نحدد نوع البايانات فإذا أردنا أن نتحقق مثلا من أن القيمة فى صندوق النص ١ أكبر من القيمة فى صندوق النص ٢ ولم نحدد نوع البيانات فى الخصيصة type سنحصل على نفس النتيجة.

تنويه : بهذا الإعتقاد قد أخطأت..

جرب معى تطبيق فيه صندوقين نص وأداة CompareValidator لها الخصائص التالية :

⊞ Font	
ForeColor	Red
Text	**textbox2 is less than textbox1**
⊟ **Behavior**	
ControlToCompare	**TextBox1**
ControlToValidate	**TextBox2**
CultureInvariantVa	False
EnableClientScript	True
Enabled	True
EnableTheming	True
EnableViewState	True
Operator	**GreaterThan**
SetFocusOnError	False
SkinID	
ToolTip	
Type	String
ValidationGroup	
ValueToCompare	
Visible	True

تلاحظ فى الصورة أننا تركنا الخاصية Type لها القيمة String مع أننا سنقارن ارقام Integersوهذا سينتج عنه خطأ فظيع .. فقط تابع.

تذكر أن الهدف من تطبيقنا هو التحقق من أن القيمة فى صندوق النص ٢ أكبر من القيمة فى صندوق النص ١.

الآن نفذ مشروعك واكتب فى صندوق النص الأول القيمة "٩" وصندوق النص الثانى القيمة "٨" الآن سيظهر خطأ على الأداة لأن القيمة فى الصندوق الثانى أقل من القيمة فى الصندوق الأول و هذا الطبيعى

الآن اكتب فى صندوق النص الأول القيمة "١٠" وصندوق النص الثانى القيمة "٩" ماذا تتوقع ؟ !

تتوقع ظهور رسالة خطأ على الأداة .. تفكيرك منطقى وسليم ولكن سيحدث عكس هذا ..حيث سيعبر التحقق بسلام ولن تظهر أى مشكلة..

السبب هو أنك حددت القيمة **string** للخاصية **type** للأداة حيث ستعامل أداة التحقق القيم فى صناديق النصوص على انها حروف فستقرأ أداة التحقق القيم من اليسار وتقارن قيمة بقية وستجد أن التسعة أكبر من الواحد فتعتبر أن القيمة فى الصندوق الثانى أكبر منها فى الصندوق الأول.

لذا احذر وتعامل مع الخاصية **type** وانت تعلم قيمتها ومن يعلم اذا كنت فى المثال السابق تتعامل مع عملات ..

--الخصيصة **ValueToCompare** وتستخدمها اذا كنت ستحدد قيمية ثابته تقرنها باقيمة التى تحملها الأداة المرتبطة بها.

--الخصيصة **ControlToCompare** والتى ستحدد الأداة التى ستتم مقارنة قيمتها بالقيمة التى تحملها الادة المرتبطة بها...

وتستخدم واحدة منهما فقط

--الخصيصة **Operator** والتى تحدد نوع المقارنة بين القيمتين وممكن تكون **DataTypeCheck Equal, NotEqual, GreaterThan, GreaterThanEqual, LessThan, LessThanEqual, and**

القيمة **DataTypeCheck** لهذه الخصيصة تجبر الأداة على التحقق اذا كان نوع البيانات المدخل فى الأداة المرتبطة بها هو نفسة نوع البيانات المحدد فى الخصيصة **Type** وإلا تفشل عملية التحقق

وكمثال على الأداة:

```
<asp:CompareValidator runat="server" Display="dynamic"
ControlToValidate="Age" ValueToCompare="18"
ErrorMessage="You must be at least 18 years old"
Type="Integer" Operator="GreaterThanEqual">*
</asp:CompareValidator>
```

يتحقق من أن القيمة التى تحملها الأداة المرتبطه بها أكبر من أو تساوى ١٨وإلا يظهر فشل فى العملية.

مثال آخر:

```
<asp:CompareValidator runat="server"
ControlToValidate="Password2" ControlToCompare="Password"
ErrorMessage="The passwords don't match"
Type="String" Display="dynamic">
</asp:CompareValidator>
```

يتحقق من أن القيمة التى يحملها صندوق كلمة المرور الأول يساوى صندوق كلمة المرور الثانى.

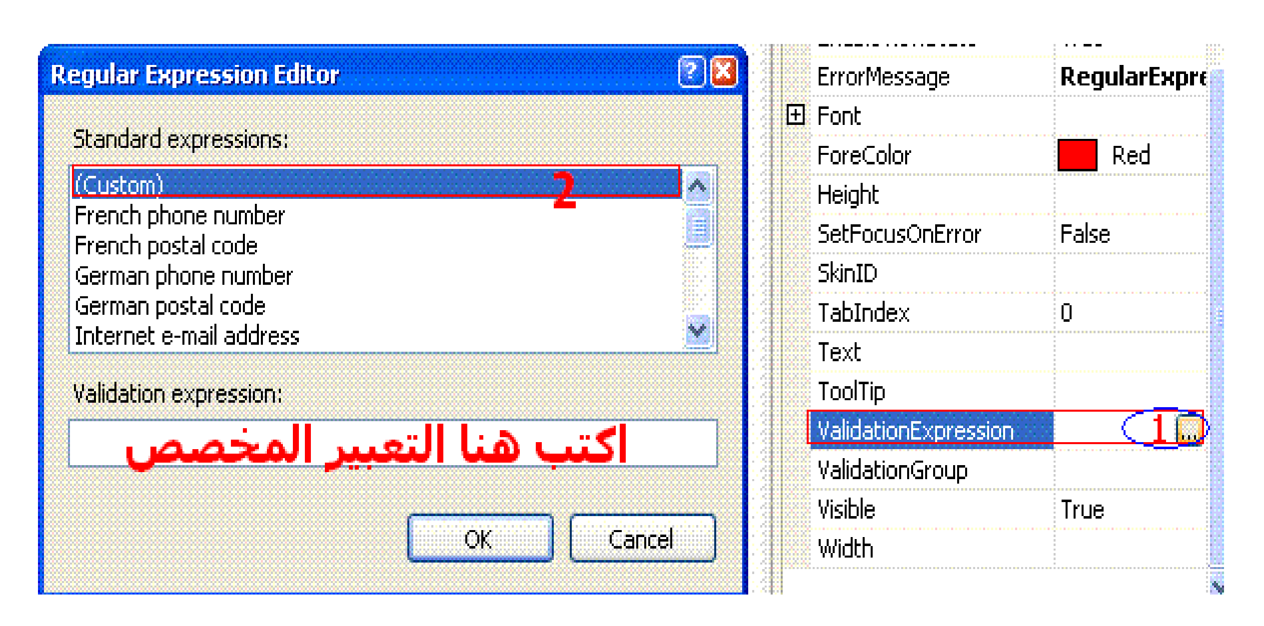
رابعا الأداة: RegularExpressionValidator Control

هذه الأداة تعد أقوى الأدةات حيث تسمح لك بالتحقق من ان البيانات التى تحملها أداة معينه تتبع صيغة معينه بريد إلكترونى أو عنوان موقع URL أو رقم تليفون وغيره.

بالرغم من انه يمكنها أن تتأكد أن المدخل هو URL أى انها تبدأ ب HTTPوينهى ب Com. مثلا إلا أنه لم يدعم البحث عن هذا العنوان على الانترنت والتحقق إذا كان موجود أم لا.

من أهم خصائصها الخصيصةValidationExpression ولها العديد من القيم مثل Internet E-mail Address:أو Internet URL وغيرها.

كما يمكنك أن تكتب التعبير يدويا عن طريق .. تابع الصورة...



فى صندوق النص الموضع بالصورة السابقة اكتب التعبير المطلوب.. ماذا ستكتب؟. ستكتب رموز تعبير بها عن التعبير المطلوب تسمى هذه الرموز MetaCharacters تراها بشكل مفصل فى هذه الروابط ..حيث أنها علم آخر ليس هذا مجالها.

خامسا الأداة: CustomValidator Control

لو عملية التحقق المطلوبة تحتاج مرونة أكبر وإمكانيات اكبر من التى تتيحها لك الأداوت السابقة(كأن تريد أن تتحقق أن القيمة المخلة هى سنة كبيسة مثلا.. Leap Year فى هذه الحالة ستلجأ للأداة. CustomValidator

بواسطة هذه الأداة يمكنك أن تكتب كود التحقق على خادم والسيرفر يدويا ..إذا فشلت عملية التحقق ستعود الطريقة IsValid() والخاصة بالفورم بالقيمة False وتظهر رسالة خطأ على الاداة مثل أى أداة أخرى.

لإجراء عملية التحقق على جهاز المستخدم عليك كتابة الأكواد بواسطة JScript وللأسف لم أتعلم الجافا سكريبت حتى الآن

ولإجراء عملية التحقق على السيرفر نأخذ المثال التالى .. حيث نريد أن نتحقق من أن القيمة التى يحملها صندوق النص المسمى txtleapyear يحمل رقم سنة كبيسة.

سنكتب أكوادنا فى الحدث CustomValidator1_ServerValidate لأداة التحقق حيث نختبر القيمة التى تحملها الأداة المرتبطه بها عن طريق فحص القيمة فى الخاصية args.valueفإذا كانت تقبل القسمة على ٤ بدون باقى ستجعل الخاصية args.isvalid تساوى true وتمر عملية التحقق بسلام.

واذا لم تكن تقبل القسمة على 4 تكون السنة غير كبيسة ونسند القيمة false للخاصة args.isvalidوسيظهر الخطأ على أداة التحقق وكذلك ستعود الطريقة page.isvalid() بالقيمة. false

الكود للحدث ServerValidate لأداة التحقق سيكون مثل التالى:

```
Dim i As Integer = CInt(args.Value)
If (i Mod 4) = 0 Then

args.IsValid = True

Else

args.IsValid = False

End If
```

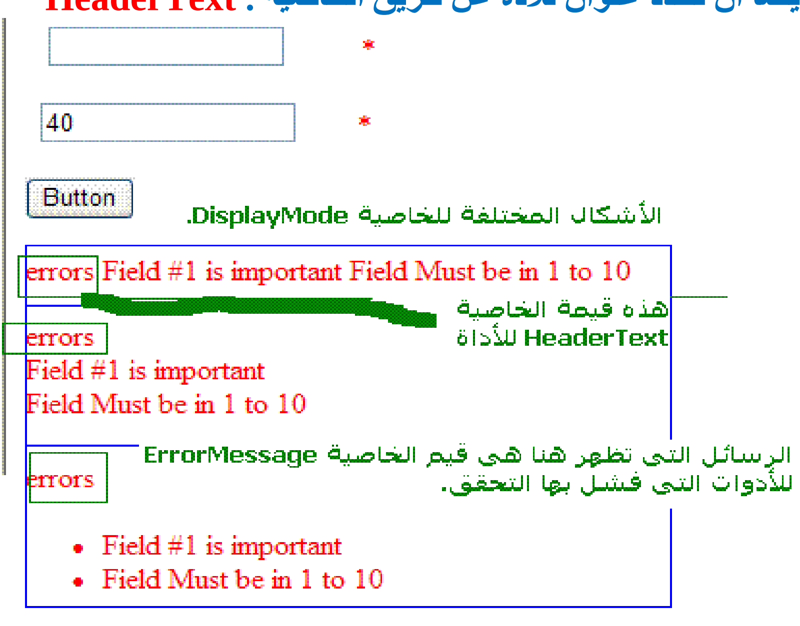
الآن انتهينا من أدوات التحقق .. تبقى لنا الأدة ValidationSummary والتى لا تعد أداة تحقق بل هى أداة لنظهر عليها كل عمليات التحقق الفاشلة فى مكان واحد وبشكل محدد.

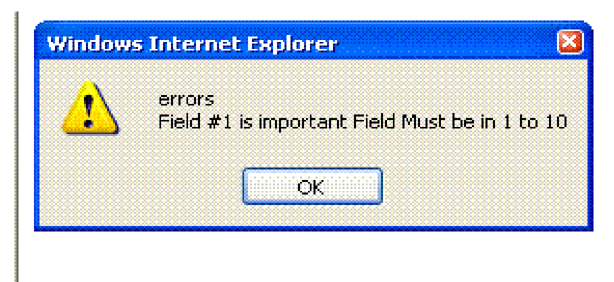
خامسا الأداة: ValidationSummary Control

وكما أشرت فى المشاركة السابقة إنها ليست أداة تحقق مثل أخواتها أى أنك لن تربطها بأداة معينة للتحقق من شىء ما ! وبدلا من ذلك فإنها تظهر ملخص لكل الأخطاء التى حدثت على الصفحة . حيث يظهر عليها القيم الموجودة فى الخاصية ErrorMessage لجميع الأدوات التى فشلت بها عملية التحقق .ويمكنك أن تخصصها ل ValidationGroupمحدد.

هذه الأداة يمكن أن تظهر على شكل رسالة (messageBox) وذلك بالاعتماد على كود (Jscript)الكود لن تكتبه أنت حيث أنها ستظهر بدون عمل PostBack لو الخاصية ShowMessageBox لها القيمة. true

كما أنها تظهر على الصفحة نفسها إذا كانت الخاصية ShowSummary لها القيمة. true
ويمكنك التعامل مع الإسلوبين أو التعامل مع إسلوب واحد فقط.
لو كان الاسلوب الثانى إحدى خياراتك : فيمكنك أن تختار الشكل الذى ستعرض به الأداة عن طريق :الخاصية : DisplayMode والتى يمكنها أن تكون SingleParagraph أو List أو.BulletList
يمكنك أيضا ان تحدد عنوان للأدة عن طريق الخاصية . HeaderText





نرى فى الصورة ظهور رسالة الخطأ على كلا من اداة التحقق نفسها وكذلك الأداة ValidationSummaryوأيضا فى messagebox

واذا اردنا ان تظهر الرسائل على الأداة ValidationSummary فقط.. نجعل الخاصية Displayلأدوات التحقق الموجودة على الصفحة تساوى. None
الآن انتهينا من الأدوات وتبقى بعض المواضيع المتفرقة

التعامل مع دوات التحقق برمجيا (عن طريق الكود)

مثل جميع الأدوات الأخرى التى تعمل على السيرفر يمكننا التعامل مع ادوات التحقق عن طريق الكود وتعديل خصائصها .. يمكنك أن تصل إلى جميع أدوات التحقق على الصفحة عن طريق ال Collectionالسمى Validators داخل الفئة page فمثلا هذا الكود يوقف عمل جميع أدوات التحقق الموجودة على الصفحة:
:

```
For Each ctrl As BaseValidator In Page.Validators

ctrl.Enabled = False

Next
```

بفرض : أنه لديك صفحة فيها صندوقى نص وأداتى تحقق وتريد أن تجعل خلفية صندوق النص الذى فشلت فيه عملية التحقق هى اللون الأحمر مثلا ..كل شىء بالكود..

أولا لابد أن يكون هناك زر أمر على الصفحة ولكن دعنا نجع الخاصية CausesValidation له تساوى false حتى لا تتم عملية التحقق آليا لأنه إذا تم التحقق آليا لن يتم ال postback وبالتالى لن ينفذ الكود فى حدث النقر وهذا غير مطلوب.

بعد ذلك اكتب فى حث النقر للزر كود كالتالى:

```
Me.Validate()
For Each ctrl As BaseValidator In Me.Validators

If ctrl.IsValid = False Then

Dim X As TextBox =
CType(Page.FindControl(ctrl.ControlToValidate), TextBox)
X.BackColor = Drawing.Color.Red
End If

Next
```

فى السطر الأول : جعلنا الصفحة تتحقق من الأدوات الموجودة عليها.

فى السطر الثانى : نبحث عن جميع أدوات التحقق الموجودة على الصفحة ويشير المتغير ctrl الى كل أدة يجدها.
فى السطر الثالث : نختبر القيمة التى تعود بها الوظيفة
IsValid()الخاصة بكل أداة على الصفحة فإذا كانت عملية التحقق فاشلة أى أن الوظيفة تعود بالقيمة>> false

فى السطر الرابع: عرفت متغير بالإسم X من النوع TextBox وبعد العلامة "=" استخدم كود مركب شوية .. تابع معى:

الوظيفة FindControl الخاصة بالصفحة Page تأخذ بارمتر عبارة عن الإسم البرمجى ID لأى أداة موجودة بالصفحة وتعود بكائن من نوع Control وهو عبارة عن الأداة الموجودة على الصفحة والتى تحمل نفس الإسم الذى مررته إليها .وبما أنها تعود بكائن من نوع Controlأى أنه ليس محدد .. استعملنا المعامل CType لتحويل الكائن من النوع Controlإلى Textbox وهكذا اسندناه إلى المتغير X والذى له النوع textBox ايضا.الآن الكائن X يشير إلى أداة صندوق النص المرتبطة بأداة التحقق التى يشير إليها المتغير. Ctrl
فى السطر الخامس : غيرنا لون الخلفية للأدة X والتى بدورها تغير لون الخلفية لصندوق النص إلى اللون الأحمر.

Validation Groups :

فى الصفحات المعقدة والكبيرة.. ربما تجد نفسك تريد أن تقوم بكل عملية تحقق لمجموعة من الأداوت على حدى .. ربما تكون الأدوات مقسمه بواسطة Panels مثلا.

كمثال على ذلك .. تريد إنشاء صفحة تحتوى على أداوت لتسجيل الدخول للموقع LogIn وفى نفس الصفحة أدوات للتسجيل فى الموقع..

وهنا تجد زر submit لكل جزء على حدى .. وطبعا بناء على كل زر على حدى تريد إجراء عملية تحقق .. فكيف تتصرف؟؟!

الحل يكمن فى تقنية وفرتها لك ميكروسوفت ..Validation Groups حيث تضع كل مجموعة من الأدوات فى ValidationGroup هذا التقسيم ليس فيزيائى أى أنك إذا وضعت كل مجموعة من صناديق النصوص وزر ال submit فى أداة panel مثلا.. لن يجدى هذا نفعا .. ولكن التقسيم يكون عن طريق الخاصية ValidationGroup لكلا من الزر ومجموعة الأدوات التى يرتبط عمله بها حيث تحصل كل مجموعة على اسم واحد خاص بها .. فأدوات تسجيل الدخول للموقع مثلا تكون الخاصية ValidationGroup لها تساوى login ومجموة الأداوت الخاصة بالتسجيل فى الموقع لها الخاصية ValidationGroup تساوى reg مثلا ...
الآن لو وضعت زر submit له الخاصية ValidationGroup فارغة هل سيترتب على ضغطة إجراء عملية تحقق لجميع الأدوات على الصفحة بغض النظر عن ال ValidationGroupالتى تنتمى إليه الادوات.

للأسف ذهب بك هذ التفكير بعيدا .. فهذا الزر لن يترتب على ضغطة إلا إجراء التحقق للأدوات التى لها الخاصية ValidationGroup فارغة أيضا وليس كل الأداوت على النموذج.
وبناء على السطر السابق ربما تقول .. لو قسمت الأدوات على الصفحة إلى Validation Groupsووضعت على الصفحة زر submit لا يتبع ValidationGroup محدد سيتم عمل postback للصفحة بنجاح وكأنه لا توجد أدوات تحقق على الصفحة أصلا ..وهذا بالتأكيد غير مطلوب.
كلامك صحيح تماما.. ولحل مشكلة مشابهة يكون الحل بإحدى الطريقتين التاليتين:.

إما أن تضاعف أدوات التحقق على الصفحة مثلا بدل من أداة تحقق RangeValidator واحدة ستضطر تضع اثنتين واحدة تتبع ValidationGroup محدد والأخرى تكون الخاصية ValidationGroupلها فارغة وذلك لمواجهة إحتمالين وهما إما أن يضغط المستخم زر submitالخاص بجموعة محددة أو يضغط الزر الذى لا يخص مجموعة محددة .. وهذا للأسف حل معقد ويفقد البرمجة بعض جمالها وبعيد عن الحياة العملية.

الحل الثانى وهو الأفضل .. أن تقسم الأدوات إلى مجموعات بدون مشاكل وكل مجموعة من الأدوات لها الخاصية ValidationGroup متشابهة .هكذا نواجه احتمال ضغط المستخدم لزر submit الخاص بمجموعة محددة.

ولمواجهة احتمال ضغط المستخدم لزر submit الذى لا يتبع مجموعة محددة نكتب فيه الكود يدويا ..وكما ناقشنا ذلك فى الجزء((التعامل مع أدوات التحقق برمجيا))

حيث يمكنك أن تستدعى الوظيفة Page.Validate() مع تحديد بارمتر نصى لها يشير إلى إسم ال validationgroupوبعد
ذلك ترى القيمة التى تعود بها الطريقة page.isvalid بعد كل.

```
page.validate("groupname")
```

ملحوظة:- الوظيفة Page.Validate() معمول لها إعادة تحميل OverLoading عند تعريفها بحيث يمكنك أن تحدد التعامل برمجيا مع validatiogroup محدد بتمرير اسمه إلها كالتالى مثلا.

```
Page.Validate("group1")
```

لاحظ أيضا مع الإهتمام : لو كانت الصفحة مقسمةإلى مجموعات وإختبرت الوظيفة isvalid () لها ستعود بالقيمة true دائما إلا اذا كانت عليها مجموعة من الأدوات التى لا تتبع مجموعة محددة .. واذا اختبرت كل مجموعة فيها على حدى مثلاpage.validate ("group١") سيكون هناك إحتما أن تعود الوظيفة page.isvalid ب true أو false حسب عملية التحقق
.
انتهينا من جميع المواضيع المختصة بأدوات التحقق .(Validation Controls) الآن
وسوف أضيف ان شاء الامثلة العملية في وقت لاحق أن شاء الله
وأعلم أن الموضوع به تقصير كبير .. لو اكتشفت خطأ راسلنى على الايميل الموجود في بداية الكتاب لتصحيحة.
وفي الاخير.......
أسال الله العظيم رب العرش الكريم أن ينصر أخواننا في غزة الحبيبة
. أمين اللهم أمين
ونسأل الله التوفيق وأرجوا من القارى الكريم نشر هذين الموقعين:-
١ـموقع غزة تحترق
٢ـصيد الفوائد
والدال على الخيركا فاعله......